



Н.С. Кравець<sup>1</sup>, М.С. Чернишов<sup>2</sup>

<sup>1</sup>Кандидат технічних наук, доцент кафедри Програмної інженерії,  
Харківський національний університет радіоелектроніки,  
natalia.kravets@nure.ua, ORCID iD: 0000-0002-6753-3333

<sup>2</sup>Студент кафедри Програмної інженерії,  
Харківський національний університет радіоелектроніки,  
mykhailo.chernyshov1@nure.ua, ORCID iD: 0000-0003-1974-730X

## ПРОБЛЕМИ РОЗГОРТАННЯ ТА КЕРУВАННЯ МУЛЬТИХМАРНИМИ РІШЕННЯМИ

Проаналізовано доцільність розгортання програмних продуктів у мультихмарі, визначено переваги та недоліки такого підходу, проблеми, пов'язані із використанням сервісів, що надаються кількома хмарними провайдерами. Досліджено, які хмарні рішення є на сьогодні найпопулярнішими, та розглянуто випадки, у яких краще використовувати саме їх, а не конкурентів. Розглянуто платформи мультихмарного управління, орієнтовані на використання організаціями з великою кількістю територіально рознесених відділів та офісів. Визначено, які фреймворки оркестрування хмарних ресурсів підтримують роботу із кількома хмарами, та за допомогою яких технологій ними здійснюється оптимальний вибір хмарного рішення.

МУЛЬТИХМАРА, ХМАРНИЙ ПРОВАЙДЕР, ФРЕЙМВОРК ОРКЕСТРУВАННЯ ХМАРНИХ РЕСУРСІВ, ПЛАТФОРМА МУЛЬТИХМАРНОГО КЕРУВАННЯ

**Кравець Н.С., Чернишов М.С. Проблемы развертывания и управления мультиоблачными решениями.** Проанализирована целесообразность развертывания программных продуктов в мультиоблаке, определены преимущества и недостатки такого подхода, проблемы, связанные с использованием сервисов, предоставляемых несколькими облачными провайдерами. Исследовано, какие облачные решения являются сегодня самыми популярными, и рассмотрены случаи, в которых лучше использовать именно их, а не конкурентов. Рассмотрены платформы мультиоблачного управления, ориентированные на использование организациями с большим количеством территориально разнесенных отделов и офисов. Определено, какие фреймворки оркестрации облачных ресурсов поддерживают работу с несколькими облаками, и с помощью каких технологий ими осуществляется оптимальный выбор облачного решения.

МУЛЬТИБЛАКО, ОБЛАЧНЫЙ ПРОВАЙДЕР, ФРЕЙМВОРК ОРКЕСТРАЦИИ ОБЛАЧНЫХ РЕСУРСОВ, ПЛАТФОРМА МУЛЬТИБЛАЧНОГО УПРАВЛЕНИЯ

**Kravets N.S., Chernyshov M.S. Problems of multi-cloud solutions deploying and managing.** The expediency of deploying software products in a multi-cloud is analyzed, the advantages and disadvantages of this approach, the problems associated with the use of services by several cloud providers are identified. It was investigated which cloud solutions are the most popular today and the cases in which it is better to use them, rather than competitors are considered. Multi-cloud management platforms are considered, which are focused on the use of organizations with a large number of geographically dispersed departments and offices. It has been determined what cloud resource orchestration frameworks support working with multiple clouds, and which technologies they use to make the best choice of a cloud solution.

MULTI-CLOUD, CLOUD PROVIDER, CLOUD RESOURCE ORCHESTRATION FRAMEWORK, MULTI-CLOUD CONTROL PLATFORM

### Вступ

Надзвичайної популярності на сьогоднішній день набули хмарні технології, які дозволяють ефективно вирішувати проблеми, що постають перед компаніями, котрим необхідно розгорнути та швидко запустити програмні продукти. Такі рішення часто дозволяють значно заощадити кошти, сплачуючи лише за використані ресурси в тому обсязі, у якому їх необхідно задіяти, а також надають можливість автоматичного масштабування на вимогу використовуваних ресурсів у реальному часі при зміні навантаження.

Зараз існує велика кількість хмарних провайдерів, що надають різноманітні послуги у хмарі: розгортання веб-застосунків, машинне навчання, штучний інтелект, сховища даних тощо. Найбільшу частку на цьому ринку мають Amazon Web Services, Microsoft Azure та Google Cloud Platform.

Варто зазначити, що кожна із хмар має власний

- набір послуг, що надаються,
- розцінки,
- поширеність у різних регіонах світу.

Тому досить часто при виборі провайдера доводилося знаходити компроміс — яка сукупність характеристик є найбільш прийнятною. Проте, насправді, немає необхідності миритися, наприклад, із відносно високою ціною орендованих віртуальних машин одного із провайдерів, якщо він при цьому має сервіси, що дозволяють значно швидше за конкурентів обробляти великі дані, котрі надходять із розумних пристроїв компанії. Оптимальним рішенням є взаємодія із різними хмарами, при чому із використанням у кожній з них тих сервісів, які найкраще підходять для вирішення конкретної задачі клієнта. Даний підхід отримав назву «мультихмара».

Мультихмари можуть спростити доступ до даних і їх аналіз, скоротити витрати, підвищити безпеку даних. Але розгортання як і керування рішенням у декількох хмарах набагато складніше організувати.

### 1. Аналіз ефективності використання мультихмари

За даними компанії Gartner [1], 81% компаній, які наразі використовують хмарні рішення, працюють із мінімум двома провайдерами одночасно. Згідно із дослідженням IDC, до 2018 року мультихмарну стратегію планували розвивати 85% ІТ-відділів великих організацій. Слід зазначити, що до 2015 року цей показник складав 10%. Аналітики IDC описують мультихмару як «організаційну стратегію або архітектурний підхід до проектування складної цифрової послуги, яка включає в себе споживання хмарних послуг від більш ніж одного постачальника хмарних послуг». Це можуть бути безпосередньо конкуруючі хмарні сервіси, такі як сховище даних від декількох постачальників загальнодоступних хмарних послуг або IaaS і SaaS від одного або декількох постачальників хмарних послуг. [2] Використання мультихмари обмежується вартістю та складністю, пов'язаною із забезпеченням узгодженого керування та управління множиною різних варіантів хмарних обчислень. Згідно даних IDC 2020 року 37 відсотків організацій використовують в ІТ-інфраструктурі мультихмарні рішення (рис. 1). [2]

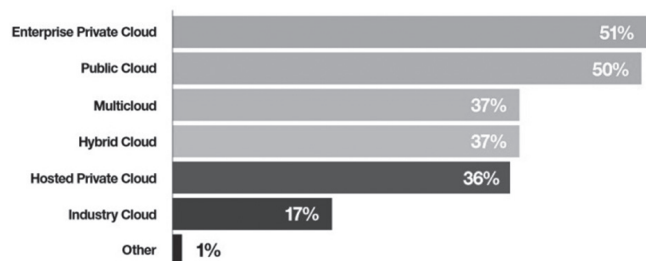


Рис. 1. Використання різних типів хмар в ІТ-інфраструктурі організацій

Мультихмарні рішення усе частіше використовуються у випадках, коли не вистачає власних потужностей для повноцінного функціонування програмної системи, а розгортання рішення у одній публічній хмарі не є можливим.

Важливою перевагою такого підходу також є незалежність від єдиного провайдера. Банкрутство такого постачальника послуг чи блокування його роботи із політичної чи інших причин, «гра не за правилами» у разі стику власних інтересів із інтересами хмарного провайдера можуть мати катастрофічні наслідки для клієнта.

Тому навіть використання аналогічних за функціоналом сервісів різних постачальників послуг у різних регіонах може виявитися ефективним — у разі припинення роботи одного із них, клієнти веб-ресурсу зможуть використовувати потужності іншого, хоч і з дещо більшою затримкою.

### 2. Використання сервісів кількох хмарних провайдерів на практиці

До того ж, немає універсальної хмари, яка може задовольнити потреби кожного із клієнтів. Для великої кількості дійсно великих багатофункціональних проектів мультихмарний підхід є єдино можливим. Наведемо приклад. Великий інтернет-магазин, серверна частина якого написана із використанням веб-фреймворку Django, який традиційно гарно інтегрується із СКБД PostgreSQL, і база даних якого вже містить велику кількість даних (тому міграція до нового рішення для підтримки сумісності призведе до переписування невиправдано великих обсягів коду) прагне впровадити найсучасніші технології штучного інтелекту і стати першим застосунком свого типу, що змінить формат взаємодії користувача із інтернет-магазином. Планується проводити верифікацію користувачів за допомогою голосу, здійснювати взаємодію із клієнтом таким чином, наче він спілкується із людиною. Традиційний підхід до роботи із сайтом також має зазнати змін: при вводі запитів до поля пошуку має використовуватися автодоповнення для зменшення кількості дій, що має виконувати користувач, проводитиметься аналіз та виправлення помилок, аби упевнитися у тому, що користувач дійсно знайде те, що мав намір придбати. Так як сайт уже є надзвичайно популярним, то виправданною буде підтримка максимально можливої кількості мов для охоплення якомога більшої аудиторії, при чому взаємодія із системою відбуватиметься у найбільш зручному для них форматі.

Використання однієї хмари у даному випадку є недоцільним. Якщо немає принципової різниці, де розгортати серверну частину за стосунку (хоча, насправді, найзручнішим і найуніверсальнішим рішенням є Amazon EC2), то для розміщення у хмарі бази даних Postgres варто використовувати Microsoft Azure, адже аналоги, наприклад, від Amazon та Google, не надають таких послуг. Коли мова йде про використання штучного інтелекту, то розглянемо рішення від Amazon, Microsoft, Google та IBM. Верифікацію за голосом підтримують лише перші два із них, найбільшу кількість мов дозволяє розпізнавати GoogleCloudAPI, автодоповнення вводу підтримується виключно Azure, а виявлення синтаксичних помилок не надається AWS.

Із проаналізованих вище рішень видно, що є такі задачі, що підтримуються лише частиною хмарних провайдерів, або ж взагалі одним. Варто зазначити, що у вибірці були найвідоміші із них, коли мова йде про більш дрібних представників, то ситуація є ще набагато складнішою. Хоча більшість знайдених рішень для даної проблеми підтримує Azure, проте не для всіх із них дане рішення є оптимальним, тому варто частину сервісів розгорнути із його використанням, а при реалізації інших — звернути увагу на більш конкурентноспроможні аналоги.

Отже, на прикладі було обґрунтовано ефективність та доцільність використання саме мультимарного підходу. Але окрім вказаних вище переваг, такий підхід містить один найважливіший недолік — складність реалізації. Так як мультимарність передбачає об'єднання в одну систему сервісів від різних платформ, то не усі із них можуть взаємодіяти один із одним, а значить, і бути інтегрованими в єдину систему. Кожна платформа має власні стандарти. Крім того, використання різних хмар може збільшити комунікаційні затримки, адже доводиться звертатися до кожної із них окремо, що, зрозуміло, відбуватиметься довше, ніж взаємодія із ресурсом у тій самій хмарі. Це означає, що більша швидкодія іншого рішення може бути зведена нанівець затримками комп'ютерної мережі. Різні платформи також надають різні рішення із забезпечення безпеки, тому необхідно визначити, чи не буде система вразливою при використанні сукупності різнорідних рішень. Більшість інструментів розгортання, налаштувань та викликів до застосунків є розрізненими, такими, що досить складно поєднуються, тому фахівцям у галузі інформаційних технологій доводиться великі проміжки часу виділяти для узгодження таких технологій [3].

Таким чином, використання сервісів низки хмарних провайдерів одночасно в рамках однієї системи дозволяє підвищити її ефективність або взагалі робить можливим її функціонування у спроектованому вигляді, проте застосовувати такий підхід потрібно лише тоді, коли це дійсно необхідно, адже інтеграція кількох різнорідних компонентів може призвести до підвищення складності програмної системи в цілому, а також до її вразливості.

### 3. Порівняння рішень найпопулярніших хмарних провайдерів

Порівняємо переваги популярних публічних хмарних платформ від Amazon, Microsoft, Google та Oracle і таким чином визначимо типи задач, що найефективніше вирішуються кожною із них, на основі чого можна вирішити, яка комбінація хмар є оптимальною для кожного окремо узятим проекту.

Коли мова йде про AWS, то дана платформа пропонує найбільшу кількість хмарних продуктів, так як вона є найстарішою і активно розвивається, пропонуючи клієнтам нові рішення. Обчислювальні сервіси є гарно масштабованими, мають високий рівень безпеки, працюючи у повністю захищеному мережевому середовищі. Серед недоліків варто зазначити досить високу ціну за послуги, що надаються, AWS Glacier має порівняно простішу функціональність, ніж інструменти резервного копіювання інших платформ.

Microsoft Azure гарно оптимізований для вирішення задач, пов'язаних із машинним навчанням та штучним інтелектом. Сервіс Data Lake має багато додаткових функцій для аналітики великих обсягів даних.

Обчислювальні сервери Google Cloud мають більш прозору у порівнянні із аналогами посекундну оплату оренди обчислювальних потужностей, інфраструктура є більш екологічно безпечною та забезпечується ощадливіше використання електроенергії у центрах обробки даних.

Oracle поки надає доступ до відносно малої кількості сервісів, проте такі рішення є досить дешевими, якщо порівнювати, наприклад, із пропозиціями Amazon. До того ж, дана хмара відома надзвичайно оптимальним автономним рішенням DBaaS (база даних як сервіс). База даних Oracle є більш досконалим рішенням, має об'єктно-реляційну модель, що дозволяє зберігати складні бізнес-дані.

AWS, Oracle, GoogleCloudPlatform (GCP) забезпечують автоматичне керування версіями, що наразі не підтримується Azure. AWS та Azure мають велику кількість інструментів моніторингу та керування, в той час коли GCP та Oracle не можуть цим похизуватися [4].

Важливою категорією порівняння хмар є кількість центрів обробки даних (ЦОД) та їх поширеність у різних регіонах планети. Це характеризує швидкість, із якою клієнти зможуть взаємодіяти із платформою, затримки в мережі. З цієї точки зору були проаналізовані хмари AWS, Alibaba Cloud, Azure та GCP. Alibaba Cloud має дуже велику концентрацію data-центрів на Дальньому Сході, насамперед у Китаї, та в Океанії, Azure — у Європі, на півдні Азії та в Австралії, а GoogleCloud та AWS — в Північній Америці. Таким чином, якщо розгортати програмні продукти у зазначених вище регіонах у відповідних хмарах, це буде оптимальним рішенням для обслуговування цільової аудиторії, що там знаходиться [5].

Відомо, що великої популярності зараз набуває машинне навчання. Саме тому його засоби намагаються інтегрувати як у велику кількість вже давно функціонуючих програмних систем, так і у такі, що лише проєктуються. Відповідно, користуються попитом і хмарні сервіси, що надають відповідні інструменти. Було розглянуто та проаналізовано рішення машинного навчання чотирьох хмар: AWS, Microsoft Azure, GCP та IBM cloud. У таблиці 1 наведено порівняння ML-сервісів даних платформ.

Таблиця 1  
Порівняння сервісів машинного навчання

	AWS	Azure	GCP	IBM cloud
Класифікація	+	+	+	+
Регресія	+	+	+	+
Кластеризація	+	+	+	-
Виявлення аномалій	+	+	-	-
Розмітка даних	+	+	+	+
Надання рекомендацій	+	+	+	-
Ранжування	+	+	-	-
Вбудовані алгоритми	+	+	+	-

Як видно із таблиці, усі основні види послуг, що були виділені для розгляду, підтримуються AWS та Azure, а в IBM-cloud відсутня підтримка більшості з них. 2021 року компанією Amazon було представлено SageMakerStudio — інтегроване середовище розробки для машинного навчання. Цей інструмент дозволяє інтегрувати усі наявні у даній хмарі засоби цієї галузі. Подібні рішення мають і конкуренти: Azure Machine Learning у Microsoft, Google Cloud AutoML у Google, IBM Watson Machine Learning Studio у IBM. Також — крім GCP, Keras — крім Azure. У AWS можна використовувати MXNet, Gluon, Chainer, Torch, у Azure — Spark ML, Microsoft Cognitive Toolkit, у GCP — XGBoost, у IBM-cloud — Spark MLib, XGBoost, PMML, IBM SPSS.

Також було проаналізовано програмні інтерфейси, призначені для обробки людського мовлення у текстовому та аудіоформаті. Результати порівняння наведено у таблиці 2.

Таблиця 2

Порівняння API машинного навчання

	AWS	Azure	GCP	IBM cloud
Розпізнавання мовлення	+	+	+	+
Перетворення тексту в мовлення	+	+	+	+
Виділення ключових фраз	+	+	+	+
Перевірка орфографії	-	+	-	-
Автодоповнення	-	+	-	-
Верифікація за голосом	+	+	-	-
Виділення метаданих	-	-	-	+
Аналіз відношень	-	+	-	+
Синтаксичний аналіз	-	+	+	+
Аналіз особистості	-	-	-	+
Обробка аудіо низької якості	+	+	+	+
Фільтрація неприпустимого вмісту	-	+	+	-
Аналіз настрою	+	+	+	+
Тегування частин мовлення	-	+	+	-
Інструменти для розробки чат-ботів	+	+	+	+

Отримані дані ще раз підтверджують тезу про те, що немає такої хмари, котра підтримувала б усі рішення, які можуть знадобитися при реалізації проекту, і говорять про необхідність використання мультихмари у разі такої потреби. За даними таблиці, Azure пропонує найбільшу кількість інструментів, проте для виділення метаданих та аналізу особистості доведеться використовувати рішення іншої хмари, із розглянутих провайдерів таке рішення має лише IBM.

Також ризикують можливості застосовувати розпізнавання мовлення, що надаються провайдерами-лідерами хмарного ринку (Таблиця 3), тому варто

ознайомитися із списком мов, що підтримуються кожним рішенням [6].

Таблиця 3

Підтримка хмарними провайдерами розпізнавання мовлення

	AWS	Azure	GCP	IBM cloud
Кількість підтримуваних мов	100+	120	120+	60+
Кількість підтримуваних мов для перекладу	6	60+	100+	48

#### 4. Платформи мультихмарного керування

Коли було здійснено вибір засобів, що використовуватимуться із кожної хмари в межах мультихмари, необхідно подумати про їх узгодження. AWS, Microsoft Azure, GCP та IBM cloud надають можливість створення розподіленого мультихмарного додатка на основі мікросервісів.

У результаті дослідження ринку, було розглянуто декілька платформ мультихмарного керування.

Перша із них — платформа мультихмарного керування від IBM, яка забезпечує централізоване керування хмарами і послугами на підприємстві. Це надає можливість використовувати різні хмари та узгоджено керувати ними в середовищі самообслуговування та співробітництва. Платформа містить у своєму складі інструмент IBM Multicloud Manager [7], він дозволяє контролювати різні кластери Kubernetes як у приватній, так і у загальнодоступній хмарі. Він надає такі зручні інструменти, як єдиний інтерфейс користувача, що дозволяє керувати мультихмарним середовищем в цілому, неперервний контроль за змінами, відкрите середовище, що унеможливує прив'язку до певного провайдера хмарних послуг. Серед аналогів його вирізняє простота використання, дана платформа чудово підходить для невеликих команд із офісами по всьому світу.

Іншим відомим рішенням є Scalr [8], який розроблений із врахуванням масштабів підприємств. Воно допомагає стандартизувати використання, контролювати витрати та дозволяє користувачам обирати найкращі хмарні сервіси для власних потреб без прив'язки до постачальника. Scalr дозволяє створювати власні політики, що можна налаштувати, які роблять можливим створення сумісного мультихмарного середовища шляхом автоматизації цих політик. Гарно підходить для великих організацій, яким необхідно керувати великою кількістю відділів та офісів. Відомими клієнтами є Xerox, Samsung, NASA JPL.

Платформа керування хмарами Bunnyshell дозволяє автоматично налаштувати, масштабувати та розгортати застосунки у декількох хмарних провайдерах. Є можливість перенесення застосунку із однієї хмари у іншу за запитом, а також моделювання міграцій.

## 5. Платформи оркестрування хмарних ресурсів

Більшість сучасних провайдерів мають власні платформи окрестрування хмарних ресурсів, які застосовуються для керування життєвим циклом ресурсів, починаючи від фази вибору і закінчуючи фазою моніторингу. Але ці продукти найчастіше є пропрієтарними та через комерційні причини не можуть бути портативними. Хоча й існують сучасні рішення для керування конфігурацією (Puppet, Chef, Amazon OpsWorks[10]), які забезпечують підтримку обробки конфігурації ресурсів через хмарні сервіси, проте часто виявляється, що користувачам необхідно розуміти низькорівневі програмні інтерфейси хмарних сервісів для того, щоб створювати та підтримувати складні конфігурації ресурсів.

Поява мультыхмарних обчислень ще більше ускладнила складні самі по собі проблеми окрестрування, проте великим платформам окрестрування довелося впроваджувати мультыхмарну підтримку, аби бути конкурентноспроможним гравцем на цьому ринку. Нові вимоги передбачають створення нових потужніших засобів окрестрування ресурсів, що охоплюють декілька хмарних адміністративних доменів, здатні впоратися з неоднорідністю базових хмарних ресурсів та сервісів. Оскільки мультыхмара передбачає відсутність попередніх домовленостей між постачальниками хмарних послуг, то доводиться користуватися послугами хмарного брокера, який є посередником між постачальниками та споживачами хмарних обчислень.

У [9] запропоновано еталонну архітектуру для фреймворків окрестрування хмарних ресурсів (ФОХР). Складається вона із ряду рівнів. Рівень доступу відповідає за регулювання взаємодії із платформою. Користувачі можуть отримувати доступ до служб із нижніх рівнів за допомогою інтерфейсів командного рядка, веб-API та панелей моніторингу. Рівень керування застосунками контролює роботу застосунків протягом усього життєвого циклу. Рівень керування ресурсами включає у себе служби (наприклад, служби виявлення, моніторингу), які керують ресурсами протягом усього їх життєвого циклу. Рівень надання ресурсів відповідає за послуги, що виконують основні операції із хмарними ресурсами (створення, запуск, масштабування, зупинка, видалення). ФОХР, що підтримують мультыхмарність, мають пропонувати рівень абстракції хмари, який приховуватиме відмінності та уникатиме необхідності індивідуального налаштування постачальника.

На базі розглянутої архітектури можна виділити два види ФОХР: виробничі/комерційні та експериментальні/академічні. Перші із них використовуються у виробничому середовищі постачальниками приватної та публічної хмари, а другі функціонують у дослідницькому середовищі.

Представниками виробничих/комерційних ФОХР, які підтримують мультыхмарність, є Cloudify, Brook-

lyn, Stratos, Alien4Cloud, Terraform. [11, 12, 13, 14]

Cloudify дозволяє розгортати мультыхмарні рішення за допомогою вбудованих плагінів. Він також підтримує BYON та використовує TOSCA для сумісності та переносимості. Недоліком є те, що складні сценарії, такі як вертикальне масштабування, не підтримуються «з коробки».

Brooklyn дозволяє розгортати мультыхмарні системи в багатьох приватних та публічних хмарах. Використовується Apache jclouds[15] у якості рівня абстракції хмари для забезпечення сумісності. Переносимість досягається за рахунок механізмів повторного та сумісного використання моделі.

У платформі Apache Stratos застосунки зазвичай складаються із наборів катриджів, які представляють опис абстрактних віртуальних машин, на яких розміщуються як бізнес-сервіси, так і сервіси інфраструктури. Для підтримки кількох постачальників також використовується Apache jclouds в якості рівня абстракції хмари. Хмарні ресурси обираються вручну при налаштуванні катриджів.

У межах Alien4Cloud застосунки можна обирати лише за допомогою ручної прив'язки. Як ручні, так і автоматичні робочі процеси можуть використовувати інструменти на основі сценаріїв чи DevOps (Chef[16], Puppet) для керування життєвим циклом застосунку.

Terraform може керувати кількома хмарними провайдерами і навіть залежностями між хмарами за допомогою спеціальних плагінів, які називаються провайдерами. Доступні постачальники для контейнерів Docker і оркестровки контейнерів, а також для зовнішніх хмарних сервісів (наприклад, Amazon RDS). Хмарні ресурси вибираються вручну під час конфігурації, в той час як дії життєвого циклу можуть бути налаштовані за допомогою ініціаторів, виконують сценарії або керуючих конфігурацією (Chef, Puppet, Salt[17]).

Серед експериментальних/академічних ФОХР, які підтримують мультыхмарність, можна виділити Clouddiator, Roboconf, INDIGO-DataCloud, MiCADO, MODAClouds, SeaClouds.[18,19,20,21,22,23]

У Clouddiator присутній брокер ресурсів, який відповідає за правильний вибір пропозиції хмари в залежності від бажаних вимог та обмежень стосовно конфігурації віртуальної машини. INDIGO-DataCloud автоматично обирає та оптимізує хмарні ресурси в залежності від SLA і даних моніторингу. Рішення для керування конфігурацією, засноване на ролях Ansible, застосовується як для розгортання програми, так і для створення попередньо налаштованих образів Docker. В межах платформи MiCADO хмарні ресурси вибираються вручну при налаштуванні віртуальних машин. Життєвий цикл додатку обробляється самою MiCADO, яка використовує OpenStack і Kubernetes для керування віртуальними машинами і контейнерами відповідно. MODAClouds використовує підхід MDE для підтримки взаємодії між

хмарними провайдерами. Зокрема, MODACloudML являє собою набір розширень UML, що дозволяють розробникам моделювати мультихмарні застосунки за допомогою трьох рівнів абстракції: хмарні незалежні обчислювальні моделі, незалежні від хмарного провайдера моделі і моделі, що залежать від хмарного провайдера. Ці моделі полегшують портативність, оскільки в більшості випадків вони їх можна використовувати багаторазово. Хмарні ресурси можна автоматично обирати й оптимізувати за допомогою Venues4Clouds і SpaceDev4Clouds, а керувати ними можна або за допомогою сценаріїв оболонки, або за допомогою Puppet[9].

### Висновки

В межах даної статті розглянуто передумови виникнення мультихмари, переваги, які отримуються при використанні хмарних ресурсів декількох провайдерів, такі як розширення функціональної наповненості програмного продукту, коли єдина хмара не має відповідного функціоналу чи він є значно обмеженим, дорожчим чи повільнішим, залежність від провайдера, а також проблеми, що пов'язані із її використанням. Визначено, що головною проблемою постає складність узгодження кількох хмарних сервісів різних постачальників.

Досліджено сучасні рішення, які можуть використовуватися для керування мультихмарними продуктами. Потужними інструментами є платформи мультихмарного керування, наприклад, від IBM, рішення Scalr, Bunnyshell. Іншим видом програмних інструментів є фреймворки окрестрування хмарних ресурсів: виробничі/комерційні (Cloudify, Brooklyn, Stratos, Alien4Cloud, Terraform) та експериментальні/академічні (Clouddiator, Roboconf, INDIGO-DataCloud, MiCADO, MODAClouds, SeaClouds), які мають засоби оркестрування ресурсів, що здатні працювати із рядом хмарних сервісів одночасно для забезпечення коректної роботи застосунку. Також визначено критерії, за якими найпопулярніші постачальники хмарних послуг краще підходять для вирішення поставленої задачі (Azure для обробки великих даних, Oracle для роботи із об'єктно-реляційними базами даних, AWS для зручного швидкого розгортання серверних частин веб-сайтів).

Отже, мультихмарні обчислення мають великі перспективи розвитку, проте використовувати їх потрібно лише у випадку необхідності, аби не зіткнути-ся із зайвими витратами та втратою часу на необхідні налаштування.

### Список літератури:

[1] Smith D., Leong L. Innovation Insight for Multicloud Computing : website.URL: <https://www.gartner.com/en/documents/3994448-innovation-insight-for-multicloud-computing> (accessed:15.05.2021).

[2] Turner M. J. Worldwide Cloud System and Service Management Software Forecast Update, 2020–2024: Enterprise Invest-

ments Rebound : website.URL: <https://www.idc.com/getdoc.jsp?containerId=US47079320> (accessed:15.05.2021).

- [3] Rethink data: A Seagate technology report with research and analysis by IDC : website.URL: [https://www.seagate.com/files/www-content/our-story/rethink-data/files/Rethink\\_Data\\_Report\\_2020.pdf](https://www.seagate.com/files/www-content/our-story/rethink-data/files/Rethink_Data_Report_2020.pdf) (accessed:15.05.2021).
- [4] Aditi Rajan Khot. A Comparative Analysis of Public Cloud Platforms and Introduction of Multi-Cloud. International Journal of Innovative Science and Research Technology. 2020. Volume 5, Issue 9. P. 448–454.
- [5] Where Are The Data Centers: AWS, Alibaba Cloud, Azure, GCP. Interconnected: website.URL:<https://interconnected.blog/data-center-coverage-aws-alibaba-azure-gcp/> (accessed:15.05.2021).
- [6] Lee Y. S. Analysis on Trends of Machine Learning-as-a-Service //International Journal of Advanced Culture Technology. – 2018. – Т. 6. – №. 4. – С. 303-308.
- [7] Raj P., Raman A. Automated multi-cloud operations and container orchestration //Software-Defined Cloud Centers. – Springer, Cham, 2018. – С. 185-218.
- [8] Mohamed A. M., Abdelsalam H. M. A multicriteria optimization model for cloud service provider selection in multicloud environments //Software: Practice and Experience. – 2020. – Т. 50. – №. 6. – С. 925-947.
- [9] Tomarchio O., Calcaterra D., Di Modica G. Cloud resource orchestration in the multi-cloud landscape: a systematic review of existing frameworks //Journal of Cloud Computing. – 2020. – Т. 9. – №. 1. – С. 1-24.
- [10] Lu M. et al. An orchestration framework for a global multi-cloud //Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference. – 2018. – С. 58-62.
- [11] de Carvalho L. R., de Araujo A. P. F. Performance Comparison of Terraform and Cloudify as Multicloud Orchestrators //2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID). – IEEE, 2020. – С. 380-389.
- [12] Carrasco J. et al. Bidimensional cross-cloud management with TOSCA and Brooklyn //2016 IEEE 9th International Conference on Cloud Computing (CLOUD). – IEEE, 2016. – С. 951-955.
- [13] Son S. et al. Cloud SLA relationships in multi-cloud environment: models and practices //Proceedings of the 8th International Conference on Computer Modeling and Simulation. – 2017. – С. 1-6.
- [14] Calcaterra D. et al. A Comparison of Multi-cloud Provisioning Platforms //CLOSER. – 2019. – С. 507-514.
- [15] Toews E., Advocate D. Introduction to Apache jclouds //Apr. – 2014. – Т. 7. – С. 23.
- [16] Ebert C. et al. DevOps //Ieee Software. – 2016. – Т. 33. – №. 3. – С. 94-100.
- [17] KONG Y. et al. Multi-cloud storage system based on Android platform //Journal of Computer Applications. – 2017. – С. S1.
- [18] Baur D. A provider agnostic approach to multi-cloud orchestration. – 2018.
- [19] Pham L. M. et al. Roboconf: A hybrid cloud orchestrator to deploy complex applications //2015 IEEE 8th International Conference on Cloud Computing. – IEEE, 2015. – С. 365-372.
- [20] Salomoni D. et al. INDIGO-DataCloud: A platform to facilitate seamless access to e-infrastructures //Journal of Grid Computing. – 2018. – Т. 16. – №. 3. – С. 381-408.
- [21] Kiss T. et al. Micado—microservice-based cloud application-level dynamic orchestrator //Future Generation Computer Systems. – 2019. – Т. 94. – С. 937-946.
- [22] Di Nitto E. et al. Model-driven development and operation of multi-cloud applications: the MODAClouds approach. – Springer Nature, 2017.
- [23] Brogi A. et al. SeaClouds: an open reference architecture for multi-cloud governance //European Conference on Software Architecture. – Springer, Cham, 2016. – С. 334-338.

*Надійшла до редколегії 23.04.2021*